VI TECHNOLOGY

Arendar® 2005

Report Utility User Manual

Month 2005
12345-67

Contact

VI Technology, Inc.
3700 W. Parmer Ln. Ste. 200
Austin, TX 78727
(512) 327-3348
info@vi-tech.comwww.vi-tech.com

Copyright

Under the copyright law, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of VI Technology, Inc.

Trademarks

Arendar is a registered trademark of VI Technology, Inc. VI Technology is a trademark of VI Technology, Inc.

Products and names listed are trademarks or trade names of their respective companies.

© 2003–2004 VI Technology, Inc. All rights reserved.

Contents

About This Manual

Organization v
Conventions v
Related Documentation vi

Chapter 1
Arendar Test Data Management System

Overview 1-1
Components 1-1
Deploying Arendar 1-2
Arendar System Database 1-3
Test Structure 1-4
    Test Hierarchy 1-4

Chapter 2
Introduction to the Arendar Report Utility

Overview 2-1
User Roles 2-1
Report Utility Interface 2-2
    Main Window 2-2
    File Menu 2-2
        Run 2-3
        New 2-3
        Delete 2-3
        Edit 2-3
        Import 2-3
        Export 2-3
        Exit 2-3
    View Menu 2-3
    Tools Menu 2-4
    Plugin Manager 2-4
    Icon Manager 2-4
    Help Menu 2-5
        Contents 2-5
        About Arendar Report Utility 2-5
        About VI Technology 2-5

Chapter 3
Installation and Configuration

Installing the Report Utility 3-1
Logging On 3-1
    Database Settings 3-2

Database System Type 3-2
Database Server Name/Database Service Name 3-2
Database Catalog/Database Schema Name 3-3
Use Windows NT Integrated Security 3-3
Use Specified Username and Password 3-3
Working Folder Path 3-3
If Database is not available 3-3
Enable Use of MSMQ to Send and Receive Test Data 3-3
Use Local Intermediate Files 3-4
Write Test Data Directly to Database 3-4
Arendar Update Manager 3-4

Chapter 4
Running a Report

Chapter 5
Plugins

Overview 5-1
Hyperlink Plugin 5-1
Excel Plugin 5-2
General Tab 5-2
Details Tab 5-2
Parameters Tab 5-3
General Tab 5-3
ComboBox Tab 5-4
Queries Tab 5-6
Date Parameters 5-8
Boolean Parameters 5-8
Optional Parameters 5-9
NULL and NOT NULL Parameters 5-9
Query Testing 5-10
Actions Tab 5-11
Report Base 5-11
Excel Report 5-11
Attachments Tab 5-12

Chapter 6
Report Managers

Icon Manager 6-1
Plugin Manager 6-1

Appendix A
Troubleshooting

About This Manual

The *Arendar Administration Utility User Manual* provides information about using the Arendar Administration Utility as part of an Arendar Test Data Management (TDM) System.

This manual is intended for test system developers and administrators.

Organization

This document is organized as follows:

- Chapter 1, *Arendar Test Data Management System*, describes the components and deployment of the Arendar Test Data Management system, and the structure and function of the Arendar System Database.
- Chapter 2, *Introduction to the Arendar Report Utility*, Introduction to the Arendar Report Utility, describes the function and use of the report utility software.
- Chapter 3, *Installation and Configuration*, describes the product interface and provides instructions for installing, launching, and configuring the Report Utility, and lists database, system, and communication setting options.
- Chapter 4, *Running a Report*, describes how to specify parameters and run a report in the Arendar Report Utility.
- Chapter 5, *Plugins*, provides instructions on using the Icon Manager and the Plugin Manager to customize reports.
- Chapter 6, *Report Managers*, describes how to use the plugins that ship with the Arendar Report Utility software.
- Appendix A, *Troubleshooting*, provides contact information for VI Technology.

Conventions

The following conventions are used in this document:

- **Bold** text indicates user interface elements, menu commands, and options.
- **Bold Arial** text indicates classes, methods, properties, and objects.
- `Courier` text indicates text entered by the user, code examples, literal values, pathnames, filenames, and extensions.
- *Italic* text indicates document titles and key terms.
- `<name>` is a string that is replaced by actual values.
- `[parameter]` is a string that is an optional parameter in a command.

Related Documentation

The following documents provide additional information about the Arendar application:

- *Arendar® 2005 Developer Manual*—Provides a comprehensive description of all classes, properties and objects available for each of the supported programming languages.
- *Arendar® 2005 Data Viewer User Manual*—Describes how to use the Arendar Data Viewer for viewing and retrieving test data.

CHAPTER 1

Arendar Test Data Management System

This chapter describes the purpose, structure, and components of the Arendar test data management (TDM) software and the Arendar System Databse.

Overview

The Arendar TDM software package is a scalable set of applications for configuring, storing, retrieving, and reporting test data. Arendar has .NET, ActiveX, and COM interfaces so that you can use any development environment to manage your test data.

Components

Arendar includes the following modules and applications:

- *Administration Utility*—Configures Oracle, Microsoft SQL Server, and Microsoft SQL Server 2000 Desktop Engine (MSDE) databases for storing automated test station data.
- *Dashboard*—Runs on the Web server and provides encryption and security to access data and reports of any test station from any PC connected to the Internet.
- *Data Viewer*—Retrieves data and generates analytical charts from the database.
- *Server Module*—Interfaces the Oracle, Microsoft SQL Server, or Microsoft SQL Server 2000 Desktop Engine (MSDE) centralized database with all test stations.
- *Station Client*—Communicates between your test station and Arendar Server Module with .NET, ActiveX, and COM interfaces to send your test results into Arendar TDM system database.
- *Data Import Agent*—Runs as a background process that analyzes and imports data from legacy applications into the Arendar TDM system.
- *Data Import Wizard*—Walks the user through the process of mapping and converting legacy data files for importing into an Arendar TDM system. This application also generates that configuration files used by the Data Import Agent.

Deploying Arendar

You can deploy Arendar in either of the following environments:

- Implement Arendar in an enterprise environment, as shown in Figure 1-1, to create a TDM system with multiple test stations saving test data into a single database. You also can have multiple clients that use either the Data Viewer or the Dashboard components to review data stored in the database.
- Implement Arendar as a stand-alone system in a single test station environment, as shown in Figure 1-2, to create a self-contained system that administers the test, stores the data and results on a database located in the system, and then allows you to view the data using tools residing on that system.

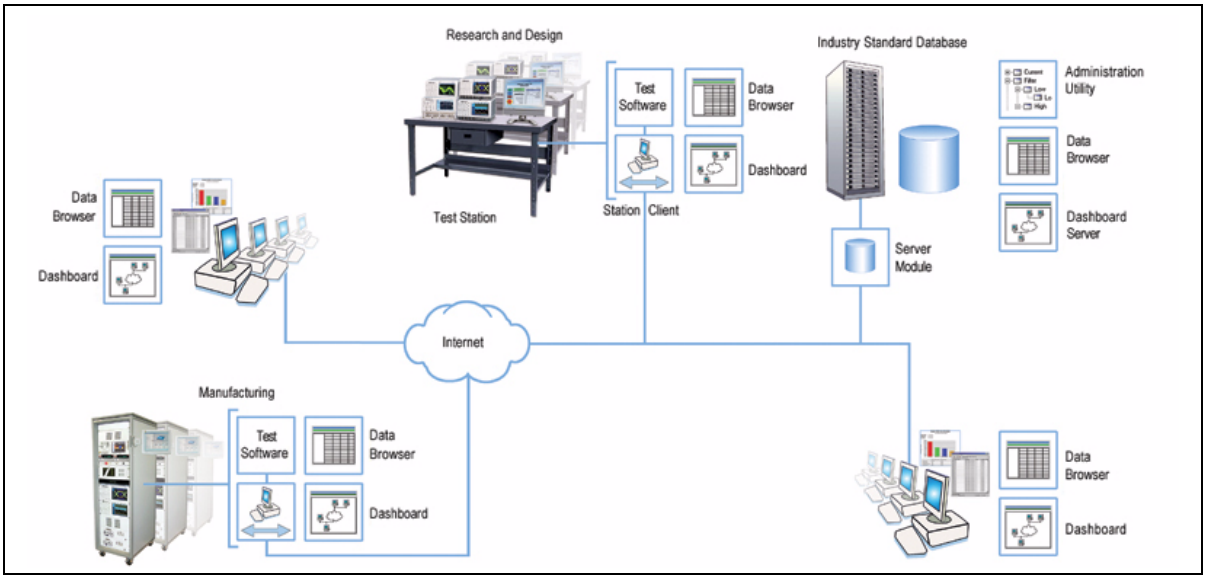


Figure 1-1. Arendar Enterprise Envorinment

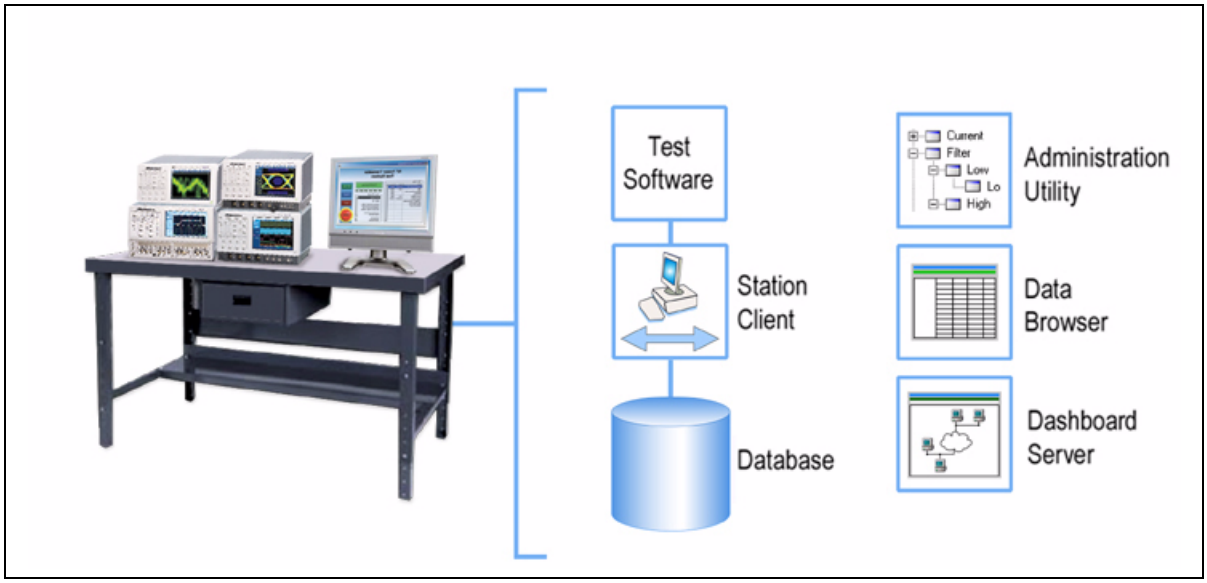


Figure 1-2. Arendar Single Test System Environment

Refer to the *Arendar 2005 Developer Manual* for information on using the Server and Station Client.

Arendar System Database

The Arendar system implements a TDM system database that stores Test Structure and Results Data, as shown in Figure 1-3.

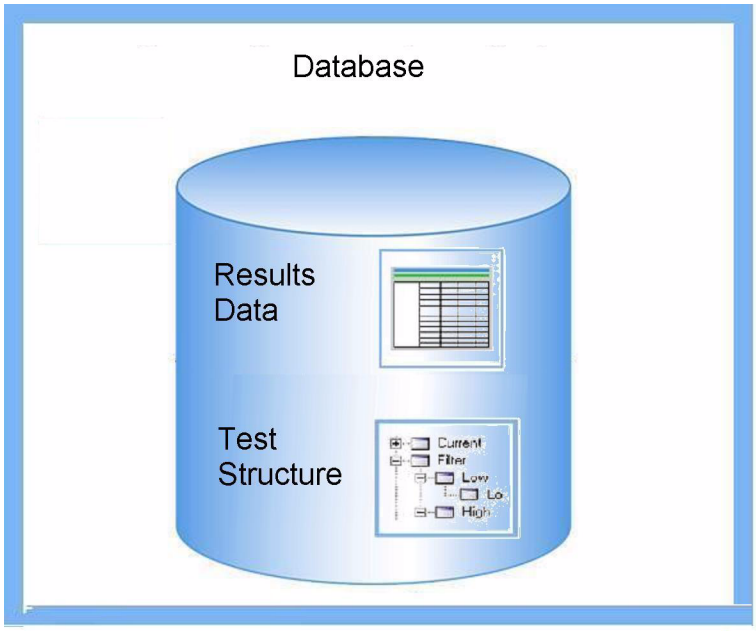


Figure 1-3. Arendar System Database

Test Structure

The Test Structure contains all structural information, including hierarchical and specific information about each *Product Test*, *Test Procedure*, and *Results Table*.

- A *Product Test* is a top-level test object that holds information specific to the product, such as the product name, part number, version, description, and pass/fail result on the test for that product. A "product" can be a physical device under test, or any other type of test that can produce a pass/fail result or return a measurement, such as measuring air quality.
- A *Test Procedure* is an object related to a product test that holds information specific to a test step or a test plan. The test procedure stores calculated, single-value results of the test, such as signal averages, signal maximum, signal minimum, dynamic range, and distortion calculations, as well as the pass/fail result of the test step or test plan.
- A *Results Table* holds one dimensional (1D) or two dimensional (2D) tabular data from your tests, such as raw waveform data measured by the test system and analyzed data array results like spectral analysis.

Test Hierarchy

Figure 1-4 shows the test hierarchy of the Sample_Product_A_Test, which is made up of the In_Process_Test and Final_Test Test Procedures, and a Waveform_Table Results Table.

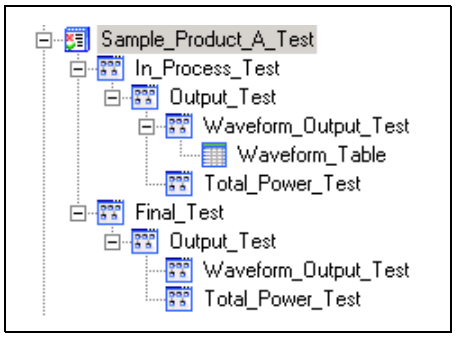
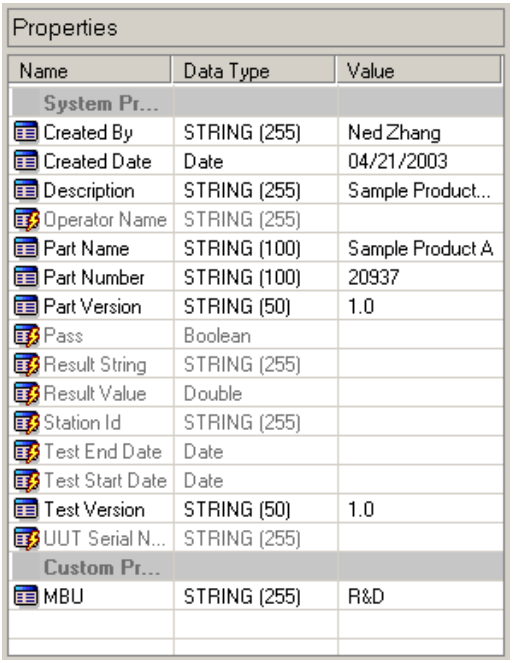


Figure 1-4. Sample_Product_A_Test Test Structure

For each Product Test and Test Procedure, the Test Structure includes *Configuration Properties*, as indicated by the icon, and *Execution Properties*, as indicated by the icon. Configuration Properties are those attributes that describe the test. They include properties such as "Test Version" or "Part Number" as shown in Figure 2-1, *Main Report Window*. Execution Properties are those attributes that are meant to store test data during an execution. "Operator Name," "Result Value," and "UUT Serial Number" are examples of Execution Properties. Their values are populated during test runtime.



Properties

Name	Data Type	Value
System Pr...		
Created By	STRING (255)	Ned Zhang
Created Date	Date	04/21/2003
Description	STRING (255)	Sample Product...
Operator Name	STRING (255)	
Part Name	STRING (100)	Sample Product A
Part Number	STRING (100)	20937
Part Version	STRING (50)	1.0
Pass	Boolean	
Result String	STRING (255)	
Result Value	Double	
Station Id	STRING (255)	
Test End Date	Date	
Test Start Date	Date	
Test Version	STRING (50)	1.0
UUT Serial N...	STRING (255)	
Custom Pr...		
MBU	STRING (255)	R&D

Figure 1-5. Execution and Configuration Properties in a Test Procedure

The Results Data portion of the Arendar system database is the information that is written to the Arendar TDM database at test runtime to store an instance of test results. You create an object representing a Product Test Execution through calls to the Arendar API.

The Product Test Execution object retrieves configuration parameter data and execution properties from the Test Structure side of the Arendar Database. During a test execution, Execution Properties and Results Tables are populated with values. The Arendar system marks each test execution with a unique execution ID.

CHAPTER 2

Introduction to the Arendar Report Utility

This chapter describes the function and user interface of the Arendar Report Utility.

Overview

The Arendar Report Utility enables Arendar users to develop and publish programmatically generated reports in a multi-user environment. The utility allows for automation of complex reports using report plugins. Plugins generate a specific kind of report, such as Excel, text, or HTML. Depending on the plugin, text, images, plots, and hyperlinks can all be incorporated into the final report. After a report has been defined by a report developer, any Arendar user can generate the report directly from the report utility or on the Web through the Arendar Web Dashboard.

The Report Utility supports importing and exporting (serializing) to XML files of all objects.

User Roles

The following permissions are granted by the Arendar Report Utility to the three classes of users.

- Report Developer—Permissions to create, edit, and delete reports and icons
- Report Administrator—Permissions to modify icons and plugins
- All Arendar users—Permissions to execute reports

Table 1. User Roles

Tasks	Report Administrator	Report Developer	All Arendar Users
Edit Plugins	Yes	—	—
Edit Icons	Yes	—	—
Edit Reports	—	Yes	—
Run Reports	Yes	Yes	Yes

Report Utility Interface

Main Window

The main report window, shown in Figure 2-1, displays all the reports available in the Arendar database.

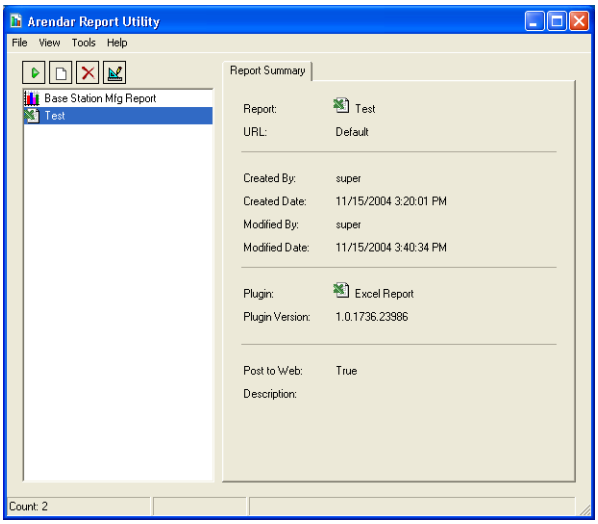


Figure 2-1. Main Report Window

The main report window contains a list of available reports on the left and a Report Summary on the right. The Report Summary includes report type, history, URL, plugin interface, and developer comments.

File Menu

From the File menu, shown in Figure 2-2, reports can be imported from and exported to XML files. You use the toolbar buttons to create, edit, delete and execute reports.

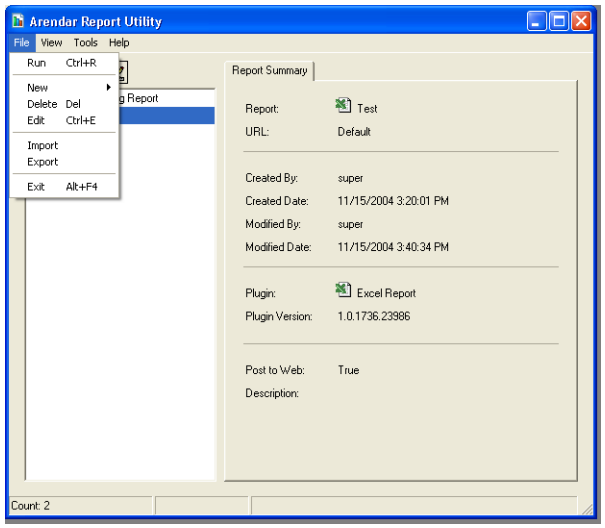


Figure 2-2. File Menu

Run

Select **Run** to launch the report selected in the report list displayed in the main window. Refer to Chapter 4, *Running a Report*, for more information.

New

Select **New** to generate a new report in the application chosen from the **New** submenu.

Delete

Select **Delete** to delete the object selected in the report list. All child objects will be deleted as well.

Edit

Select **Edit** to alter the settings of the selected report. Refer to Chapter 5, *Plugins*, and Chapter 6, *Report Managers*, for more information on editing reports.

Import

Select **Import** to add a report onto your system from an XML file.

Export

Select **Export** to automatically save all reports listed in the Report Utility in XML format.

Exit

Select **Exit** to exit the Report Utility.

View Menu

Select **View»Refresh** to refresh the list of reports in the Report Utility window and display any changes made to reports by other users on the network.

Tools Menu

The Tools menu contains two options: **Icon Manager** and **Plugin Manager**, shown in Figure 2-3.

.

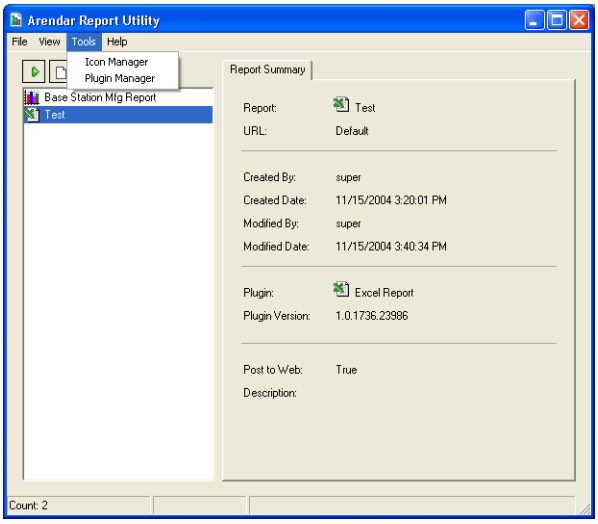


Figure 2-3. Tools Menu

Plugin Manager

The Plugin Manager allows you to change the plugin utility of a report.

Refer to Chapter 5, *Plugins*, for more information about using the Plugin Manager.

Icon Manager

The Icon Manager allows you to create, add, and change icons in your Report Utility interface.

Refer to Chapter 6, *Report Managers*, for more information about using the Icon Manager.

Help Menu

Use the Help menu, shown in Figure 2-4, to get additional information about VI Technology and the Arendar software.

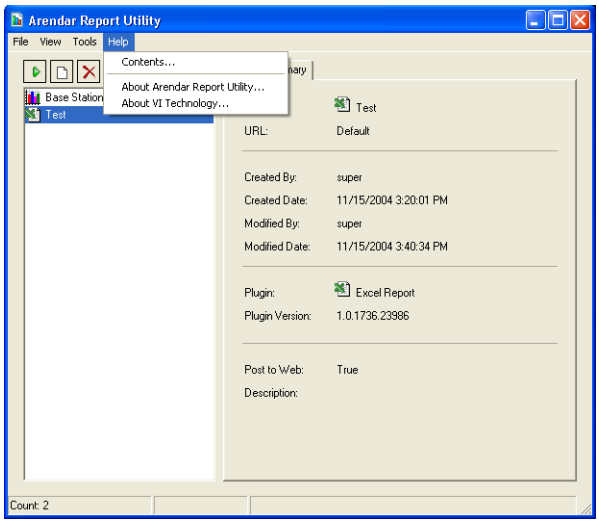


Figure 2-4. Help Menu

Contents

Select **Help»Contents** to open the HTML version of the *Arendar 2005 Report Utility User Manual*.

About Arendar Report Utility

Select **Help»About Arendar Report Utility** to display version information about the software.

About VI Technology

Select **Help»About VI Technology** to display information regarding VI Technology, Inc.

CHAPTER 3

Installation and Configuration

This chapter provides instructions for installing, launching, and configuring the Report Utility, and lists database, system, and communication setting options.

Installing the Report Utility

TBD—Installer not developed yet.

Logging On

When you first launch the Arendar Report Utility, a Logon window opens, as shown in Figure 3-1.

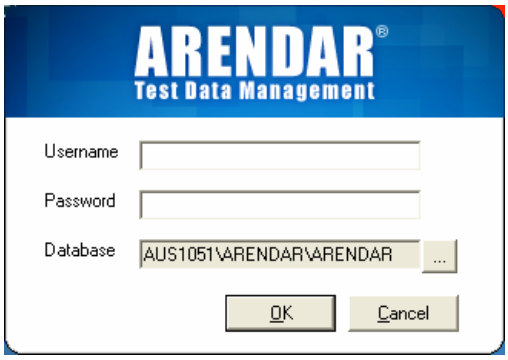


Figure 3-1. Entering User Name and Password

Contact your system administrator for your username, password, and database settings if you do not know them already.

To change the database settings, select the ... icon. The Arendar - Program Settings window opens, as shown in Figure 3-2.

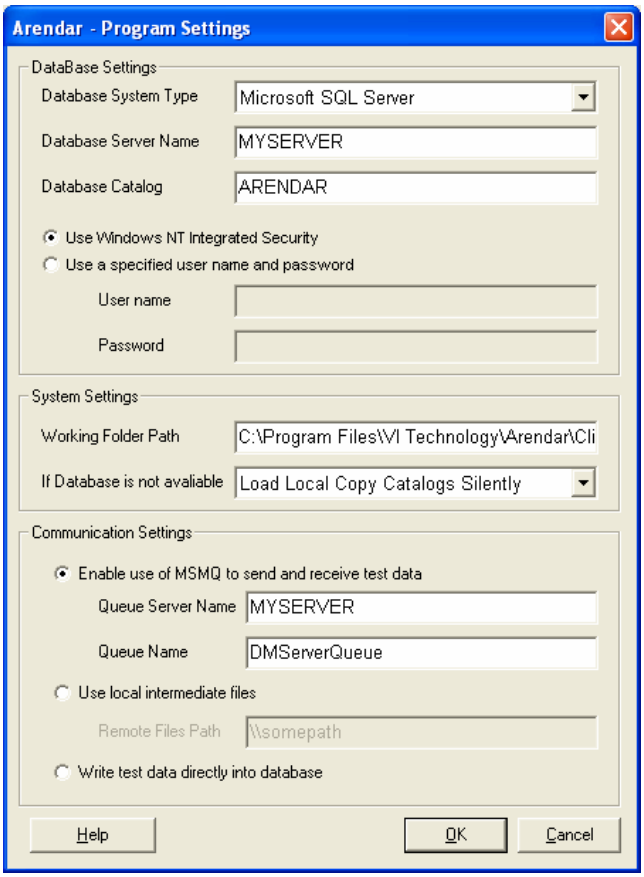


Figure 3-2. Arendar - Program Settings Window

The Arendar - Program Settings window allows you to configure the settings that determine the server and database catalog, database location, security permissions, and communication methods. The following sections describe database, system, and communication configuration options.

Database Settings

Database System Type

The *Database System Type* setting determines which database system Arendar uses. The options are Microsoft SQL Server and Oracle Database.

When using MSDE, select Microsoft SQL Server.

Database Server Name/Database Service Name

For the Microsoft SQL Server, the *Database Service Name* is the name of your database server. The name can be a partially or fully qualified Domain Name Server (DNS) name (for example, `myserver.mydomain.com`).

For an Oracle Database, the *Database Service Name* is the net service name of the Arendar database. You can configure the net services using the Oracle Net Manager utility.

Database Catalog/Database Schema Name

For the Microsoft SQL Server, the *Database Schema Name* is the name of the database catalog you are using—for example, `MyDatabase`. For an Oracle Database, this is the username (schema) under which the Arendar system stores the data.

Use Windows NT Integrated Security

This setting enables the database to use the Windows NT integrated security option. This allows the clients to connect using the Windows user that is currently logged on in that computer.

Use Specified Username and Password

This setting allows clients to connect to the database server using the specified name and password.

System Settings

Working Folder Path

The *Working Folder Path* is the full path to the folder the client uses as the main working folder. The default value is a folder named `ClientDataFolder`, located within the installation root directory.

If Database is not available

This combo box provides a set of options to load the client test definitions while the database is not accessible. The options include **Load Local Catalogs Silently**, **Load Local Catalogs Verbosely**, or **Do Not Load Local Catalogs**.

Communication Settings

Enable Use of MSMQ to Send and Receive Test Data

This setting configures the Arendar client to send data to the server using Microsoft Message Queue (MSMQ) as the transfer method. For this setting, you must install and configure the Arendar Server to support MSMQ.

- Queue Server Address—Enter the name or address of the server configured to handle MSQM. The address can be a partial or fully qualified DNS name (for example, `myserver.mydomain.com`).
- Server Queue Name—Enter the name of the MSMQ queue configured on the server. The default is `DMServerQueue`.

Use Local Intermediate Files

This setting specifies the remote location to which data are written before they are sent to the database. If the network is not available, the system ensures the transfer at a later time. After the intermediate files are transferred to the specified remote location, the server processes the files and enters the data into the database.

The *Remote Files Path* is the remote location where the intermediate files are placed for later processing by the Arendar System server.

Write Test Data Directly to Database

This setting specifies that the Arendar System writes data directly to the database. You must install either SQL Server or Oracle on the client machine.

Note This option does *not* provide protection if the network goes down.

Arendar Update Manager

After a successful logon, the Arendar Report Utility checks the database for current component versions. The software automatically downloads and installs the latest versions of sub-components if necessary.

Figure 3-3. Arendar Update Manager

CHAPTER 4

Running a Report

This chapter explains how to specify parameters and run a report in the Arendar Report Utility.

Any Arendar user can run a defined report. Highlight the report to execute and select **Run** from the File menu or click the **Run** toolbar button. Depending on the report plugin, a report configuration interface displays. If the plugin utilizes the default report configuration window, a report window with the report title, description, and optional image displays, along with any defined parameters needed to generate a report.

Figure 4-1 shows the configuration window for a Noise Test report that requires you to enter the desired serial number to report on. Required parameters have an asterisk (*) next to the parameter label. A value of <ANY> signifies no value entered for that parameter and is not a valid value for a required parameter.

After you specify all relevant parameters, click the **Create Report** button. The Report Utility generates the report based on the input parameters.

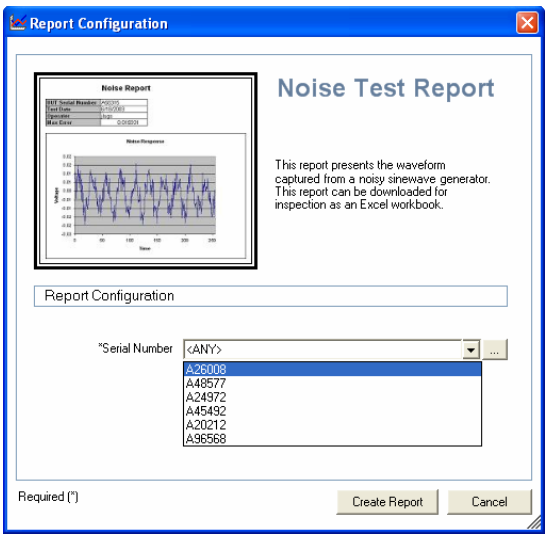


Figure 4-1. Report Configuration Window

CHAPTER 5

Plugins

This chapter describes how to use the plugins that ship with the Arendar Report Utility software.

Overview

A plugin is a .NET assembly with a public creatable class implementing the `VITechnology.Arendar.CustomReports.IPlugin` interface. Plugins allow the Arendar Report Utility to generate a number of report types. Plugins they support automatic code updates from the Arendar database and can be imported and exported to XML. You can develop custom plugins for the Arendar Report Utility to create expanded report types.

The Arendar Report Utility ships with two plugins: Hyperlink and Excel.

Hyperlink Plugin

Hyperlink reports redirect the user to the specified URL using the default browser. This is useful for directing users to a Web page with custom reporting logic.

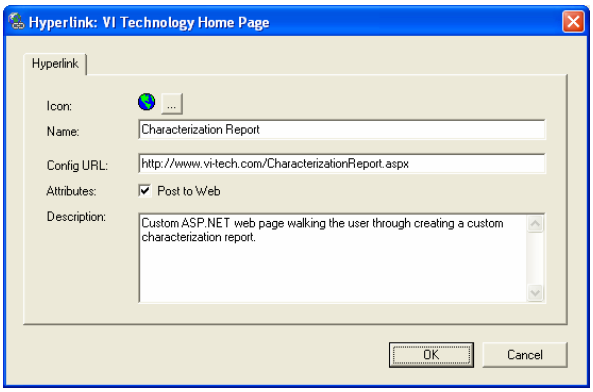


Figure 5-1. Hyperlink Report

To define a hyperlink report, you can specify an icon to display for the report, as well as a name, description, URL, and whether the hyperlink should be accessible from the Arendar Web Dashboard.

Excel Plugin

The Excel plugin allows developers to define reports using an Excel workbook.

General Tab

The General tab, shown in Figure 5-2, specifies the report Name, Description, and, optionally, an icon and image.

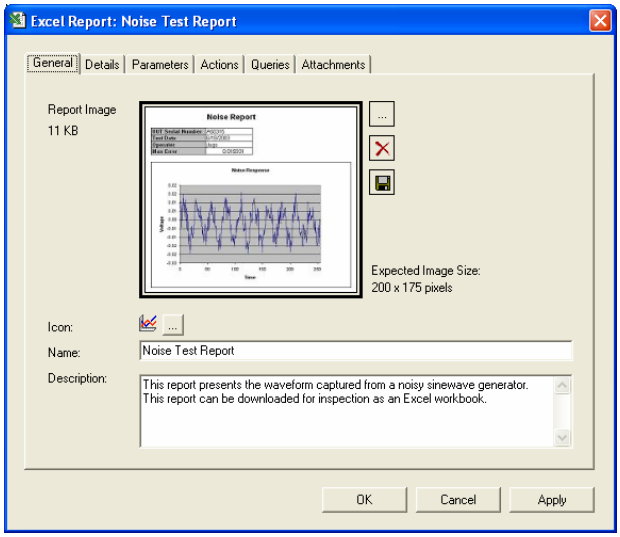


Figure 5-2. General Tab

Details Tab

The Details Tab, shown in Figure 5-3, summarizes the following report definition attributes:

- Plugin
- Plugin Version
- Created By
- Created Date
- Modified By
- Modified Date

Select **Post to Web** if you want the report to be accessible through the Arendar Web Dashboard.

Note (Optional) Selecting the default value in the Config URL field displays the default report configuration window if running from the windows Report Utility or the default configuration page in the Web dashboard. Selecting a custom URL launches a Web browser and navigates to the specified URL.

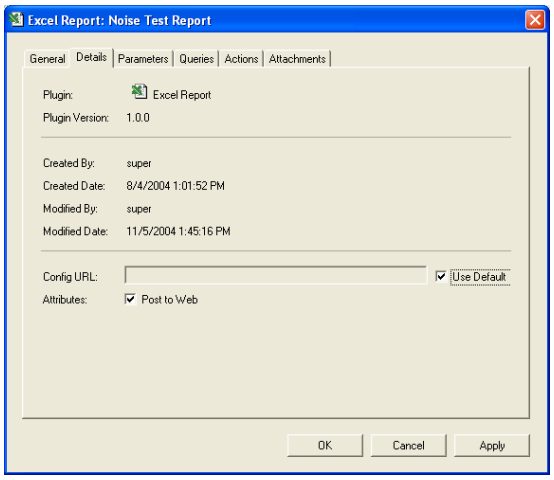


Figure 5-3. Details Tab

Parameters Tab

Use the Parameters tab, shown in Figure 5-4, to specify inputs required by the user to generate the desired report. A common example is Serial Number, where the generated report is dependent on the serial number requested.

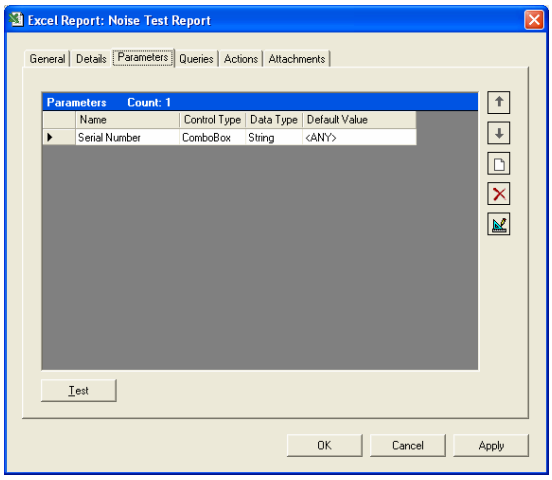


Figure 5-4. Parameters Tab

General Tab

You can display Parameters using two input control types:

- TextBox—Allows the use to enter text directly
- ComboBox—Allows the developer to display a list for the user to choose from

You can specify the parameter name, default value, and data type. Available data types are Boolean, Date, Double, Integer, and String. You can also specify whether the input parameter is required, accepts a `NULL` value, or accepts a `NOT NULL` value. `NULL` is a database field value representing an empty or null field.

A value of <ANY> specifies that the operator has chosen not to provide a value for the parameter and is not a valid input for required parameters. You can select default values of NULL, NOT NULL, <ANY>, and dates from the parameter menu button on the right of the parameter default value.

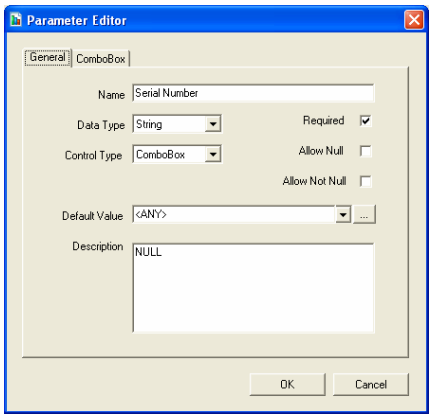


Figure 5-5. General Tab

ComboBox Tab

The ComboBox tab, shown in Figure 5-6, contains parameter attributes specific to ComboBox parameter control types. You can specify the drop down style to use with the ComboBox. The drop down styles are as follows:

- DropDown—User can select from list or enter value directly.
- DropDownList—User can select an option from the list only.

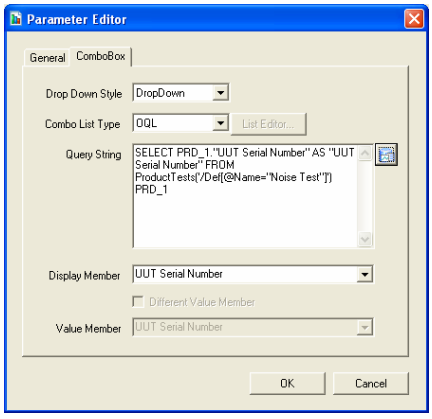


Figure 5-6. ComboBox Tab

The Combo List Types ar as followse:

- OQL—Selection list is obtained by executing an OQL query.
- SQL—Selection list is obtained by executing an SQL query.
- List—Selection list is specified by the developer.

If the list type is OQL or SQL, the Query String, Display Member, and Value Member fields are enabled. To aid in creating the query string, the Arendar Data Viewer can be launched from the Data Viewer button.

Refer to the *Arendar Data Viewer* manual for a description of Data Viewer operation.

After the query is created in the Data Viewer, you can continue editing it directly from the Query String textbox. For example, the Data Viewer might create an OQL string to query for all Serial Numbers in product “Noise Test.”

```
SELECT PRD_1."UUT Serial Number" AS "UUT Serial Number" FROM
ProductTests('/Def[@Name="Noise Test"]') PRD_1
```

A typical modification is to remove duplicate values from the returned dataset. To do this, you can directly enter the `DISTINCT` keyword to return only distinct values.

```
SELECT DISTINCT PRD_1."UUT Serial Number" AS "UUT Serial Number" FROM
ProductTests('/Def[@Name="Noise Test"]') PRD_1
```

To return only the top 25 values, enter the following:

```
SELECT DISTINCT TOP 25 PRD_1."UUT Serial Number" AS "UUT Serial Number" FROM
ProductTests('/Def[@Name="Noise Test"]') PRD_1
```

After you specify a query string, you must choose the field from the returned dataset that is to be displayed and, optionally, a different value field if the chosen value differs from the displayed value.

If the combo list type is List, the **List Editor** button is enabled. You use the List Editor dialog box to specify a list of options for the user to select from. You can optionally specify a different display member for the list than the value member.

An example of this situation is using product IDs. Perhaps you need to specify a product ID (1, 2, 3), but it is more meaningful to display a list of product names to select from (Milk, Eggs, Butter).

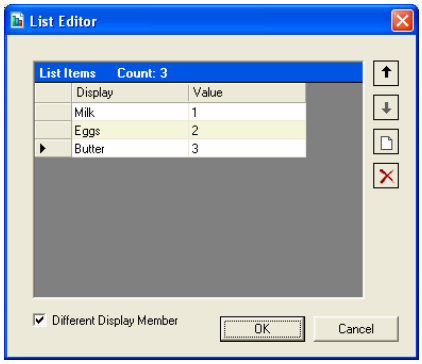


Figure 5-7. List Editor

When you are finished, click the **Test** button to see the interface presented to the user without exiting the editing window.

Note Clicking the **Close** button will *not* generate the report when testing.

Queries Tab

Use the Queries tab, shown in Figure 5-8, to create the queries needed to retrieve report data. A report can be composed of multiple queries with results dependent on the input parameters entered by the user. For example, a report query might return different results based on the value of a “Serial Number” parameter entered by a user.

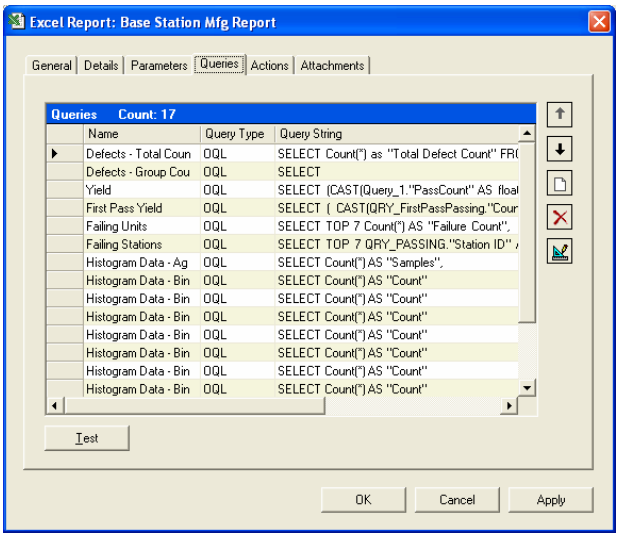


Figure 5-8. Queries Tab

The query editor dialog box, shown in Figure 5-9, allows you to specify a query name, type (OQL or SQL), and query string. To aid in creating the query string, the Arendar Data Viewer can be launched from the Data Viewer button.

Refer to the *Arendar Data Viewer* manual for a description of Data Viewer operation.

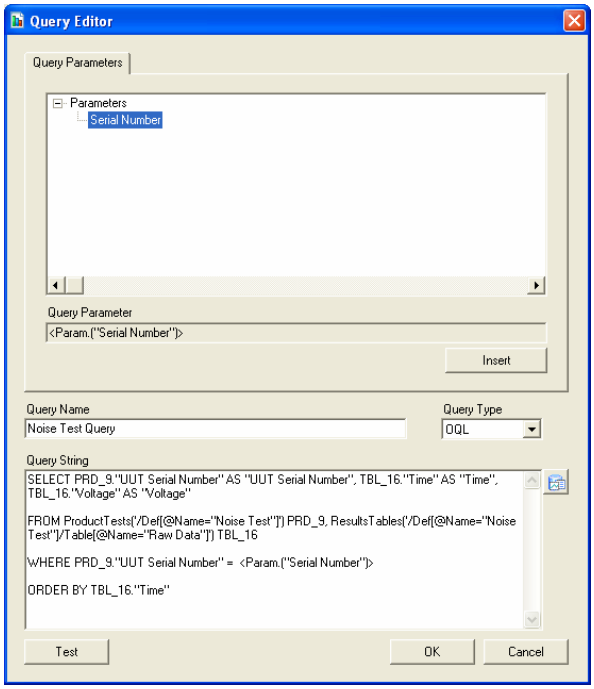


Figure 5-9. Query Editor

After the query is created in the Data Viewer, you can continue editing it directly from the Query String textbox. For example, the Data Viewer might create an OQL string to query for Serial Number, Time, and Voltage in product `"Noise Test"` where the Serial Number is `'SN001'`.

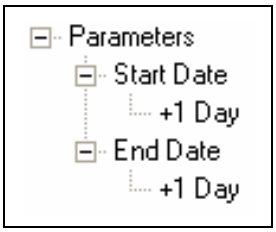
```
SELECT PRD_9."UUT Serial Number" AS "UUT Serial Number", TBL_16."Time" AS
"Time", TBL_16."Voltage" AS "Voltage"

FROM ProductTests('/Def[@Name="Noise Test"]') PRD_9,
ResultsTables('/Def[@Name="Noise Test"]/Table[@Name="Raw Data"]') TBL_16

WHERE PRD_9."UUT Serial Number" = 'SN001'

ORDER BY TBL_16."Time"
```

If you enter a desired value for the serial number filter instead of filtering by `Serial Number = 'SN001'`, you can insert the Parameter value in place of the criteria `'SN001'` by selecting **Insert** from the Parameter tree.

```
SELECT PRD_9."UUT Serial Number" AS "UUT Serial Number", TBL_16."Time" AS
"Time", TBL_16."Voltage" AS "Voltage"

FROM ProductTests('/Def[@Name="Noise Test"]') PRD_9,
ResultsTables('/Def[@Name="Noise Test"]/Table[@Name="Raw Data"]') TBL_16

WHERE PRD_9."UUT Serial Number" =  <Param.("Serial Number")>

ORDER BY TBL_16."Time"
```

Notice that the quotes surrounding `'SN001'` must be removed. The report utility will enclose the input parameter in the appropriate database delimiter for the criteria data type.

Date Parameters



Date parameters have special values of `+1 Day`. This allows for inclusive end dates. For example, suppose you specify Start Date and End Date input parameters for a report. The `WHERE` criteria in the query string might look like the following:

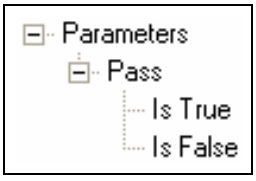
```
WHERE (PRD_3."Test Start Date" >= <Param.("Start Date")>) AND (PRD_3."Test
End Date" <= <Param.("End Date")>)
```

With the `WHERE` criteria, the end date is not inclusive. If you were to enter `1/1/1990` for the start date and `1/1/2000` for the end date, only dates up to 12/31/1999 23:59:59 would be included. The day 1/1/2000 would not be included in the returned values as you might expect. To resolve this problem, use the `+1 day` parameter in the `WHERE` string instead.

```
WHERE (PRD_3."Test Start Date" >= <Param.("Start Date")>) AND (PRD_3."Test
End Date" <= <Param.("End Date").+1Day>)
```

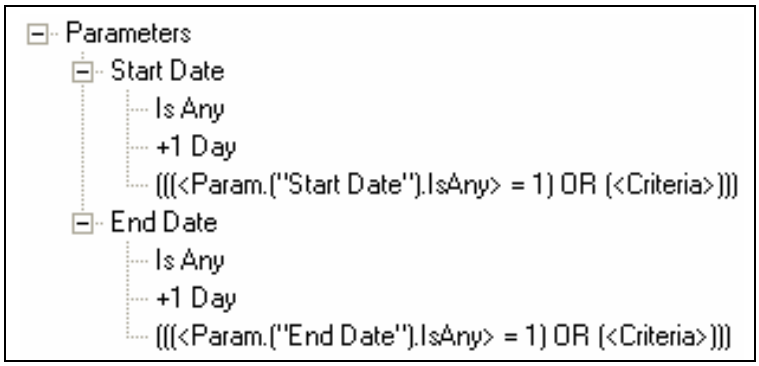
This will add one day to the End Date value you entered, effectively making the End Date inclusive.

Boolean Parameters



Boolean parameters return 1 if True, or 0 if False. Boolean parameters have special values of `Is True` and `Is False`. These parameters return 1 if the expression evaluates to True, or 0 otherwise. If you enter `False` into a Boolean parameter, the Boolean parameter returns 0, the `Is True` parameter returns 0, and the `Is False` parameter returns 1.

Optional Parameters



Optional parameters add an extra level of complexity. Suppose you specify optional Start and End dates. Both parameters have default values of `<ANY>`. Selecting `<ANY>` means there is no limit on that parameter. Having both Start Date and End Date values of `<ANY>` specifies no date restrictions on the query. The parameter tree creates additional parameters for optional inputs to help you handle optional criteria. The `Is Any` parameter returns 1 if the parameter value is `<ANY>`, or 0 otherwise.
The `(((<Param.("End Date").IsAny> = 1) OR (<Criteria>)))` string evaluates a criteria based on optional parameters. In this example, the `WHERE` criteria string would be

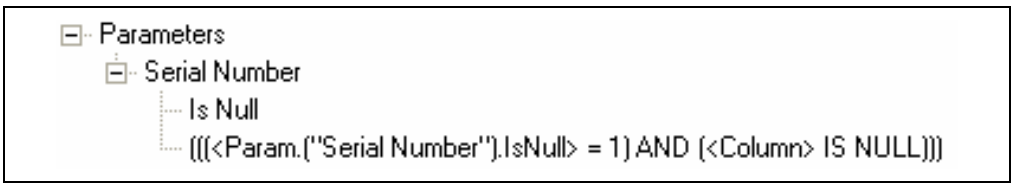
```
WHERE (((<Param.("Start Date").IsAny> = 1) OR
  (PRD_3."Test Start Date" >= <Param.("Start Date")>)))
AND
(((<Param.("End Date").IsAny> = 1) OR
  (PRD_3."Test End Date" <= <Param.("End Date").+1Day>)))
```

if the Start Date parameter is left as <ANY>, <Param.("Start Date").IsAny> = 1. In that case, the WHERE criteria

```
(((<Param.("Start Date").IsAny> = 1) OR
  (PRD_3."Test Start Date" >= <Param.("Start Date")>)))
```

would always evaluate to True because the first half of the logical expression would be (1 = 1) which is always true. When you select `<ANY>`, the first expression in the `WHERE` criteria is always short-circuited to True. If you enter a value other than `<ANY>`, the first half of the logical expression would be (0 = 1) which is always False. In that case, the `WHERE` criteria is determined by the second expression after the OR.

NULL and NOT NULL Parameters



NULL and NOT NULL parameters require additional evaluation parameters. Suppose you specify a Serial Number that can be NULL. The parameter tree creates additional parameters for NULL inputs to help you handle nullable criteria. The Is Null parameter returns 1 if the parameter value is NULL, or 0 otherwise. The (((<Param.("Serial Number").IsNull> = 1) AND (<Column> IS NULL))) string evaluates a criteria based on nullable parameters.

```
WHERE (((<Param.("Serial Number").IsNull> = 1) AND
  (PRD_9."UUT Serial Number" IS NULL)))
```

If the Serial Number parameter is specified to be NULL, the previous logical expression evaluates to True for those records where Serial Number is null. If the Serial Number parameter is NOT NULL, this logical expression always evaluates to False because the first half of the expression evaluates to (0 = 1).

Query Testing

Click the **Test** button to interactively edit and execute queries. When executed, the selected queries in the Query Plan control, shown in Figure 5-10, are executed. After completion, selecting different queries in the Query Plan control displays the query string and results for that query. You can modify all query strings directly from this dialog box.

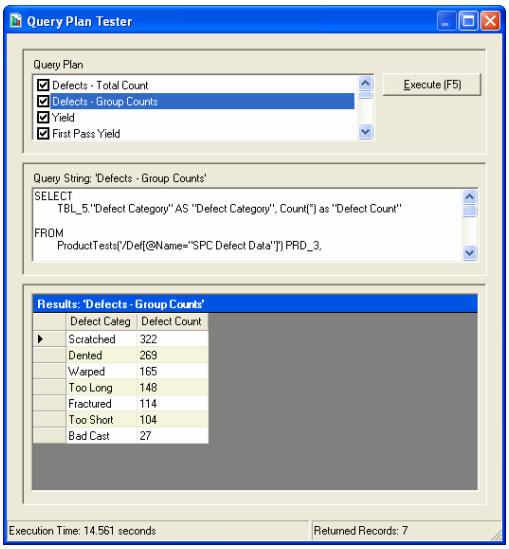


Figure 5-10. Query Plan

Actions Tab

Use the Actions tab, shown in Figure 5-11, to perform actions with the parameters and query results entered by the user.

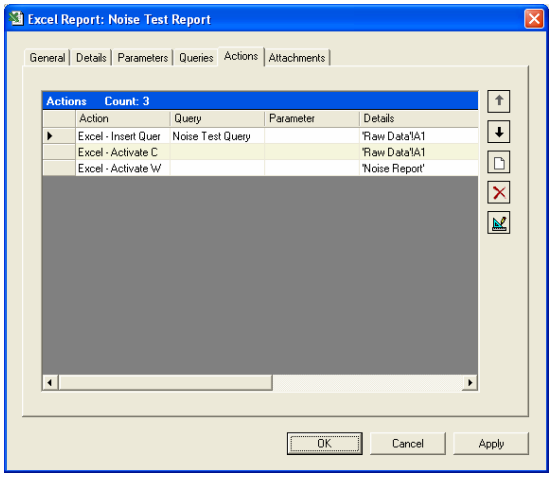


Figure 5-11. Actions Tab

For the Excel Plugin, the following actions are available.

Report Base

- Execute Query—Forces the report utility to execute a query. This is typically not required because queries are automatically executed. This action provides developers with exact control of query execution when memory management is a concern.
- Unload Query—Forces the report utility to unload query results from memory. This is typically not required because queries are automatically unloaded upon report completion. This action gives you exact control of query resources when memory management is a concern, as is the case with a report containing many large queries.

Excel Report

- Insert Query—Inserts query contents into an Excel workbook at the location specified by the worksheet and cell.
- Insert Query with Heade— Inserts query contents and header information into an Excel workbook at the location specified by the worksheet and cell.
- Insert Parameter—Inserts parameter contents entered by the user into an Excel workbook at the location specified by the worksheet and cell.
- Insert Parameter with Header—Inserts parameter contents and header information entered by the user into an Excel workbook at the location specified by the worksheet and cell.

- Execute Macro—Executes the specified macro. If the report is to be posted to the Web, macro code that displays dialog boxes, forms, or other user interface components generates an error in the Arendar Web Dashboard.
- Activate Worksheet—Activates the specified worksheet in an Excel workbook. This is typically done at the end of report generation just before saving so that the desired worksheet is activated on next open.
- Activate Cell—Activates the specified cell in an Excel worksheet. This is typically done at the end of report generation just before saving so that the desired cell is activated on next open.

Attachments Tab

Use the Attachments tab, shown in Figure 5-12, to upload an Excel workbook template (`*.xls`) to the database for use when generating the report. If left empty or cleared, the Excel plugin creates a new Excel workbook when run with default worksheets of `Sheet1`, `Sheet2`, and `Sheet3`.

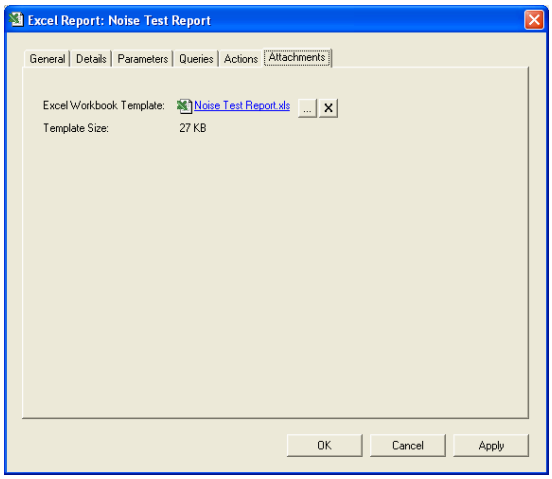


Figure 5-12. Attachments Tab

Developing an Excel template is an iterative process where the report under development is run and the created workbook is populated with data. You format the data by creating charts, tables, text, and images until an acceptable report is created. Then you delete the data inserted by the report utility, upload the final empty template into the report definition, and save.

CHAPTER 6

Report Managers

This chapter provides instructions on using the Icon Manager and the Plugin Manager to customize reports.

Icon Manager

The Icon Manager, shown in Figure 6-1, is launched from the Tools menu (Report Administrators only). Icons may be created, edited, and deleted. All edits are final. Icons may be imported and exported to XML from the File menu. The icon file is saved to the database as a BLOB (Binary Large OBject), which can be downloaded for editing.

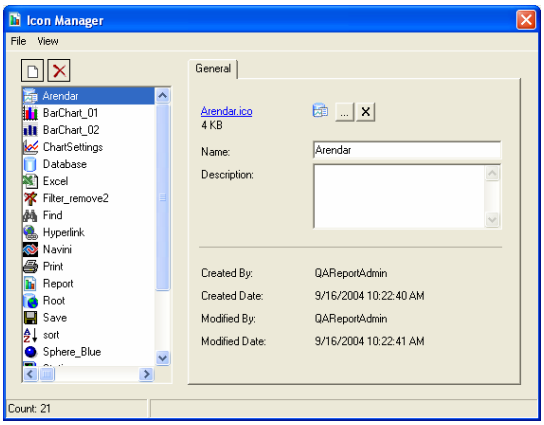


Figure 6-1. Icon Manager

Plugin Manager

The Plugin Manager, shown in Figure 6-2, is launched from the Tools menu (Report Administrators only). A plugin is a .NET assembly with a public creatable class implementing the `VITechnology.Arendar.CustomReports.IPlugin` interface. Plugins can be created, edited, and deleted. All edits are final.

Also, plugins can be imported from and exported to XML from the File menu. Plugins support automatic code updates from the Arendar database. A default icon is selectable for each plugin definition. You can develop plugins and add them to the Arendar Report Utility for custom report types.

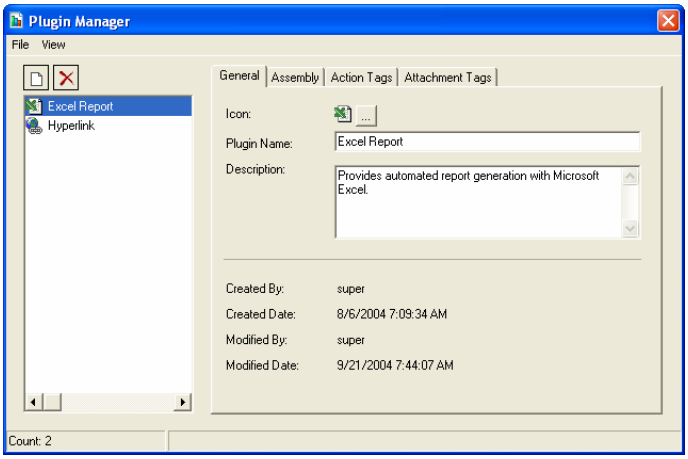


Figure 6-2. Plugin Manager

You can upload the .NET assembly containing the plugin to the database as a BLOB (Binary Large OBject). When the Report Utility requires a plugin, it loads the plugin from the local computer and checks the plugin version against the latest version in the database. If the plugin does not exist locally or the version is out of date, the report utility automatically downloads the latest assembly from the database and loads the plugin dynamically.

Plugins can also define the actions and attachments that they support. The plugin's specific actions and attachments are used during report development.

APPENDIX A

Troubleshooting

For more information about the Arendar Administration Utility, contact VI Technology by visiting `www.vi-tech.com` or by calling (512) 327-3348.

Index

A

administration utility 1-1
Arendar
    deployment 1-2
Arendar Report Utility. *See* report utility 2-1
Arendar test data management system. *See* test data management system

C

communication settings 3-3
components, test data management system. 1-1
configuration properties 1-4
customizing reports 6-1

D

dashboard 1-1
data
    import agent 1-1
    import wizard 1-1
    viewer 1-1
database settings 3-2
deploying Arendar
    enterprise environment 1-2
    stand-alone system 1-2

E

enterprise environment 1-2
execution properties 1-4

H

help, troubleshooting A-1
hyperlink report 6-1

I

icon manager 2-4, 6-1

L

logging on to report utility 3-1

P

plugin manager 2-4, 6-1
plugins
    Excel 5-2
    hyperlink 5-1
    list editor 5-5
    overview 5-1
    queries 5-6
    query editor 5-7
product test 1-4

Q

queries
    creating 5-6
    testing 5-10

R

related documentation 1-vi
report managers
    icon 6-1
    plugin 6-1
report utility
    interface 2-2
    logging on 3-1
    overview 2-1
    running report 4-1
    update manager 3-4
    user roles 2-1
results table 1-4
running report 4-1

S

server module 1-1
settings
    communication 3-3
    database 3-2
    system 3-3
stand-alone test system
    deploying 1-2
    figure 1-3
station client 1-1

system
    database (figure) 1-3
    settings 3-3

T

test data management system
    components
        administration utility 1-1
        dashboard 1-1
        data import agent 1-1
        data import wizard 1-1
        data viewer 1-1
        server module 1-1
        station client 1-1
    overview 1-1
test structure
    configuration properties 1-4
    execution properties 1-4
    product test 1-4
    results table 1-4
    test procedure 1-4
troubleshooting A-1